Epsilon-net method for optimizations over separable states

Yaoyun Shi and Xiaodi Wu
Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, USA

Abstract

We give algorithms for the optimization problem: $\max_\rho \langle Q, \rho\rangle$, where Q is a Hermitian matrix, and the variable ρ is a bipartite *separable* quantum state. This problem lies at the heart of several problems in quantum computation and information, such as the complexity of QMA(2). While the problem is NP-hard, our algorithms are better than brute force for several instances of interest. In particular, they give PSPACE upper bounds on promise problems admitting a QMA(2) protocol in which the verifier performs only logarithmic number of elementary gate on both proofs, as well as the promise problem of deciding if a bipartite local Hamiltonian has large or small ground energy. For $Q \geq 0$, our algorithm runs in time exponential in $\|Q\|_F$. While the existence of such an algorithm was first proved recently by Brandão, Christandl and Yard [*Proceedings of the 43rd annual ACM Symposium on Theory of Computation* , 343–352, 2011], our algorithm is conceptually simpler.

1 Introduction

Entanglement is an essential ingredient in many ingenious applications of quantum information processing. Understanding and exploiting entanglement remains a central theme in quantum information processing research [HHH+09]. Denote by $\mathrm{SepD}(\mathcal{A}_1 \otimes \mathcal{A}_2)$ the set of separable (i.e, unentangled) density operators over the space $\mathcal{A}_1 \otimes \mathcal{A}_2$. A fundamental question known as the *weak membership* problem for separability is to decide, given a classical description of a quantum state ρ over $\mathcal{A}_1 \otimes \mathcal{A}_2$, whether this state ρ is inside or ϵ far away in trace distance from $\mathrm{SepD}(\mathcal{A}_1 \otimes \mathcal{A}_2)$. Unfortunately this very basic problem turns out to be intractable. In 2003, Gurvits [Gur03] proved the NP-hardness of the problem when ϵ is inverse exponential in the dimension of $\mathcal{A}_1 \otimes \mathcal{A}_2$. The dependence on ϵ was later improved to inverse polynomial [Ioa07, Gha10].

In this paper we study a closely related problem to the weak membership problem discussed above. More precisely, we consider the linear optimization problem over separable states.

Problem 1. Given a Hermitian matrix Q over $\mathcal{A}_1 \otimes \mathcal{A}_2$ (of dimension $d \times d$), compute the optimum value, denoted by $\mathrm{OptSep}(Q)$, of the optimization problem

$$\max \langle Q, X\rangle \text{ subject to } X \in \mathrm{SepD}(\mathcal{A}_1 \otimes \mathcal{A}_2).$$

It is a standard fact in convex optimization [GLS93, Ioa07] that the weak membership problem and the weak linear optimization, a special case of Problem 1, over certain convex set, such as $\mathrm{SepD}(\mathcal{A}_1 \otimes \mathcal{A}_2)$, are equivalent up to polynomial loss in precision and polynomial-time overhead. Thus the hardness result on the weak membership problem for separability passes directly to Problem 1.

Besides the connection with the weak membership problem for separability, Problem 1 can also be understood from many other aspects. Firstly, as the objective function is the inner-product of a Hermitian matrix and a quantum state, which represents the average value of some physical observable, the optimal value of Problem 1 inherently possesses certain physical meaning. Secondly, in the study of the tensor product space [DF92], the value $\mathrm{OptSep}(Q)$ is precisely the *injective norm* of Q in $\mathcal{L}(\mathcal{A}_1) \otimes \mathcal{L}(\mathcal{A}_2)$, where $\mathcal{L}(\mathcal{A})$ denote the Banach space of operators on $\mathcal{A}$ with the operator norm. Finally, one may be equally motivated from the study in operations research. The definition of Problem 1 appeared in an equivalent form in [LQNY09] with the new name of "Bi-Quadratic Optimization over Unit Spheres". Subsequent works [HLZ10, So11] demonstrate that Problem 1 is just a special case of a more general class of optimization problems called homogenous polynomial optimization with quadratic constraints, which is currently an active research topic in that field.

Another motivation to study Problem 1 is the recent interest about the complexity class QMA(2). Originally the class QMA (also known as *quantum proofs*) was defined [KSV02] as the quantum counterpart of the classical complexity class NP. While the extension of NP to allow multiple provers trivially reduces to NP itself, the power of QMA(2), the extension for QMA with multiple *unentangled* provers, remains far from being well understood. The study of the multiple-prover model was initiated in [KMY01, KMY03], where QMA(k) denotes the complexity class for the k-prover case. Much attention was attracted to this model because of the discovery that NP admits *logarithmic*-size unentangled quantum proofs [BT09]. This result was surprising because single prover quantum logarithm-size proofs only characterize BQP [MW05]. It seems adding one unentangled prover increases the power of the model substantially. There are several subsequent works on refining the initial protocol either with improved completeness and soundness bounds [Bei10, ABD+09, CF11, GNN11] or with less powerful verifiers [CD10]. Recently it was proved that QMA(2)=QMA(poly) [HM10] by using the so-called *product test* protocol that determines whether a multipartite state is a product state when two copies of it are given. There is another line of research on the power of unentangled quantum proofs with restricted verifiers. Two complexity classes BellQMA and LOCCQMA, referring to the restricted verifiers that perform only nonadaptive or adaptive local measurements respectively, were defined in [ABD+09] and studied in [Bra08, BCY11]. It has been shown [BCY11] that LOCCQMA(m) is equal to QMA for constant m.

Despite much effort, no nontrivial upper bound of QMA(2) is known. The best known upper bound QMA(2)$\subseteq$NEXP follows trivially by nondeterministically guessing the two proofs. It would be surprising if QMA(2) $=$ NEXP. Thus it is reasonable to seek a better upper bound like EXP or even PSPACE. It is not hard to see that simulating QMA(2) amounts to distinguishing between two promises of $\mathrm{OptSep}(Q)$, although one has the freedom to choose the appropriate Q. Note that Problem 1 was also studied in [BCY11] for the same purpose.

Hardness result. There are several approaches to prove the hardness of Problem 1. The first is to make use of the NP-hardness of the weak membership problem and the folk theorem in convex optimization as mentioned above. However, one may directly reduce the CLIQUE problem to Problem 1 [deK08, LQNY09]. There is also a stronger hardness result [HM10] on the exact running time of algorithms solving Problem 1 conditioned on the Exponential Time Hypothesis (ETH) [IP01]. The hardness results extend naturally to the approximation version of Problem 1. It is known that $\mathrm{OptSep}(Q)$ remains to be NP-hard to compute even if inverse polynomial additive error is allowed. Nevertheless, it is wide open whether the hardness result remains if one allows even larger additive error.

From the perspective of operations research, the hardness of Problem 1 is a consequence of not being a convex optimization problem. In this case although efficient methods, compared with brute-force, for finding a local optimum usually exist, on the other hand finding the global one is fraught with difficulty. This is because one needs to enumerate all possible local optima before one can determine the global optimum in the worst case.

Our contributions. In this paper we provide efficient algorithms for Problem 1 in either time or space for several Qs of interest. As the hardness result implies that enumeration is likely to be inevitable in the worst case, our idea is to enumerate via epsilon-nets more "cleverly" with the help of certain structure of Q.

When the total number of points to enumerate is not large, one can represent and hence enumerate each point in polynomial space. If the additional computation for each point can also be done in polynomial space, one immediately gets a polynomial-space implementation for the whole algorithm by composing those two components naturally. We make use of the relation NC(poly)=PSPACE [Bor77] to obtain space-efficient implementation for the additional computation, which in our cases basically includes the following two parts. The first part helps to make sure the enumeration procedure works correctly. This is because these epsilon-nets of interest in our algorithm are not standard, additional effort is necessary to generate them. This part turns into a simple application of the so-called *multiplicative matrix weight update* (MMW) method [AHK05, WK06, Kal07] to computing a min-max form, which is known to admit efficient parallel algorithms under certain conditions. The second part contains the real computation which ,in our case, only consists of fundamental matrix operations. It is well known those operations usually admit efficient parallel algorithms [Gat93]. As a result, both parts of the additional computation admit efficient parallel algorithms, and therefore, the additional computation can be implemented in polynomial space in our case.

We summarize below the main results obtained by applying the above ideas.

1. The first property exploited is the so-called *decomposability* of Q which refers to whether Q can be decomposed in the form $Q = \sum_{i=1}^{M} Q_i^1 \otimes Q_i^2$ with small M. Note this concept is closely related to a more commonly studied concept, *tensor rank*. Intuitively, if one substitutes this Q's decomposition into $\langle Q, \rho_1 \otimes \rho_2 \rangle$ and treat $\langle Q_1^1, \rho_1 \rangle, \cdots, \langle Q_M^1, \rho_1 \rangle, \langle Q_1^2, \rho_2 \rangle, \cdots, \langle Q_M^2, \rho_2 \rangle$ as variables, the optimization problem becomes quadratic and M corresponds to the number of second-order terms in the objective function. If we plug the values of $\langle Q_1^1, \rho_1 \rangle, \cdots, \langle Q_M^1, \rho_1 \rangle$ into the objective function, then the optimization problem reduces to be a semidefinite program, and thus can be efficiently solved. Hence by enumerating all possible values of $\langle Q_1^1, \rho_1 \rangle, \cdots, \langle Q_M^1, \rho_1 \rangle$ one can efficiently solve the original problem when M is small. Since this approach naturally extends to the k-partite case for $k \geq 2$, we obtain the following general result.

Theorem 1 (Informal. See Section 3)**.** *Given any Hermitian Q and its decomposition $Q = \sum_{i=1}^{M} Q_i^1 \otimes \cdots \otimes Q_i^k$ as input, the quantity $OptSep(Q)$ can be approximated with additive error δ in quasi-polynomial time*[1] *in d and $1/\delta$ if kM is bounded by poly-logarithms of d.*

By exploiting the space-efficient algorithm design strategy above, this algorithm can also be made space-efficient. To facilitate the later applications to complexity classes, we choose the input size to be some n such that $d = \exp(\text{poly}(n))$.

Corollary 1 (Informal. See Section 3)**.** If $kM/\delta \in O(\text{poly}(n))$, the quantity $\text{OptSep}(Q)$ can be approximated with additive error δ in PSPACE.

[1] Quasi-polynomial time is upper bounded by $2^{O((\log n)^c)}$ for some fixed c, where n is the input size.

As a direct application, we prove the following variant of QMA(2) belongs to PSPACE where QMA(2)$[\mathrm{poly}(n), O(\log(n))]$ refers to the model where the verifier only performs $O(\log(n))$ elementary gates that act on both proofs at the same time and a polynomial number of other elementary gates. Note QMA(2)[poly(n),poly(n)]=QMA(2) in our notation.

Corollary 2. QMA(2)$[\mathrm{poly}(n), O(\log(n))] \subseteq$ PSPACE.

This result establishes the first PSPACE upper bound for a variant of QMA(2) where the verifier is allowed to generate some quantum entanglement between two proofs. In contrast, previous results are all about variants with nonadaptive or adaptive local measurements, such as BellQMA(2) [ABD+09, Bra08, CD10] or LOCCQMA(2) [ABD+09, BCY11].

We also study Problem 1 when Q is a local Hamiltonian over k parties. Recall that a promise version of this problem in the one party case, namely the *local-Hamiltonian problem*, is QMA-complete problem [KSV02]. Our definition extends the original local Hamiltonian problem to its k-partite version. However, as will be clear in the main section, the k-partite local Hamiltonian problem is no longer necessarily QMA(k)-complete. On the other side, our enumeration algorithm based on the decomposability of Q works extremely well in this case. As a result, we obtain the following corollary.

Corollary 3 (Informal. See Section 5)**.** Given some local Hamiltonian Q over k parties as input, $\mathrm{OptSep}(Q)$ can be approximated with additive error δ in quasi-polynomial time in $d, 1/\delta$; the k-partite local Hamiltonian problem belongs to PSPACE.

Very recently, an independent result [CS11] of us shows that the 2-partite local Hamiltonian problem defined above lies in QMA, and henceforth in PSPACE, which complements our algorithmic result.

2. The second structure made use of is the eigenspace of Q of large eigenvalues. As a result, we establish an algorithm solving Problem 1 with running time exponential in $\|Q\|_{\mathrm{F}}$.

Theorem 2 (Informal. See Section 6)**.** *For positive semidefinite Q, the quantity $\mathrm{OptSep}(Q)$ can be approximated with additive error δ in time* $\exp(O(\log(d) + \delta^{-2}\|Q\|_F^2 \ln(\|Q\|_F/\delta)))$.

A similar running time $\exp(O(\log^2(d)\delta^{-2}\|Q\|_{\mathrm{F}}^2))$ was obtained in [BCY11] using some known results in quantum information theory.(i.e., the semidefinite programming for finding symmetric extension [DPS04] and an improved quantum de Finetti-type bound.) In contrast, our algorithm only uses fundamental operations of matrices and epsilon-nets. To approximate with precision δ, it suffices to consider the eigenspace of Q of eigenvalues greater than δ whose dimension is bounded by $\|Q\|_{\mathrm{F}}^2/\delta^2$. Nevertheless, naively enumerating density operators over that subspace does not work since one cannot detect the separability of those density operators. We circumvent this difficulty by making nontrivial use of the Schmidt decomposition of bipartite pure states.

We note, however, that other results in [BCY11] do not follow from our algorithm, and our method cannot be seen as a replacement of the kernel technique therein. Furthermore, our method does not extend to the k-partite case, as there is no Schmidt decomposition in that case.

Open problems. The main open problem is whether Problem 1 admits an efficient algorithm in either time or space, when larger additive error is allowed. It is also interesting to see whether, for those Qs that come from the simulation of the complexity class QMA(2), the quantity $\mathrm{OptSep}(Q)$ can be efficiently computed.

Organizations: The rest part of this paper is organized as follows. The necessary background knowledge on the particular epsilon-nets in use is introduced in Section 2. The main algorithm based on the decomposability of Q is illustrated in Section 3. Two applications of such an algorithm is discussed immediately after; the simulation of variants of QMA(2) is discussed in Section 4 and the local Hamiltonian case is discussed in Section 5. Finally, the demonstration of an algorithm with running time exponential in $\|Q\|_{\mathrm{F}}$ for Problem 1 can be found in Section 6.

Notations: We assume familiarity with standard concepts from quantum information [NC00, KSV02, Wat08]. Particularly, our notations follow from [Wat08]. Precisely, we use $\mathcal{A}, \mathcal{B}$ to denote complex Euclidean spaces and $\mathrm{L}(\mathcal{A}), \mathrm{Herm}(\mathcal{A}), \mathrm{D}(\mathcal{A})$ to stand for the linear operators, Hermitian operators and density operators over $\mathcal{A}$ respectively. We denote the trace norm of operator Q by $\|Q\|_{\mathrm{tr}}$, i.e. $\|Q\|_{\mathrm{tr}} = \mathrm{Tr}\,(Q^*Q)^{1/2}$ where Q^* stands for the conjugate transpose of Q. The Frobenius norm is denoted by $\|Q\|_{\mathrm{F}}$ and the operator norm is denoted by $\|Q\|_{\mathrm{op}}$. The ℓ_1 norm of vector $x \in \mathbb{C}^n$ is denoted by $\|x\|_1 = \sum_{i=1}^n |x_i|$ and its ℓ_∞ norm is denoted by $\|x\|_\infty = \max_{i=1,\cdots,n} |x_i|$. We use $\|x\|$ to denote the Euclidean norm. The unit ball of $\mathbb{C}^n$ under certain norm $\|\cdot\|$ is denoted by $\mathbf{B}(\mathbb{C}^n, \|\cdot\|)$.

2 Epsilon Net

The epsilon-net (or ϵ-net) is an important concept in several mathematical topics. For our purpose, we adopt the following definition of ϵ-net.

Definition 1 (ϵ-net)**.** Let (X, d) [2] be any metric space and let $\epsilon > 0$. A subset $\mathcal{N}_\epsilon$ is called an ϵ-net of X if for each $x \in X$, there exists $y \in \mathcal{N}_\epsilon$ with $d(x, y) \leq \epsilon$.

Now we turn to the particular ϵ-net considered in this paper. Let $\mathcal{H}$ be any Hilbert space of dimension d and $\mathcal{Q} = \mathcal{Q}(M, w) = (Q_1, Q_2, \cdots Q_M)$ be a sequence of operators on $\mathcal{H}$ with $\|Q_i\|_{\mathrm{op}} \leq w$, for all i. Define the $\mathcal{Q}$-space, denoted by $\mathrm{SP}(\mathcal{Q})$, as

$$\mathrm{SP}(\mathcal{Q}) = \{(\langle Q_1, \rho\rangle, \langle Q_2, \rho\rangle, \cdots, \langle Q_M, \rho\rangle) : \rho \in \mathrm{D}(\mathcal{H})\} \subseteq \mathbb{C}^M.$$

The set is convex and compact, and a (possibly proper) subset of Raw-$(M, w) = \{(q_1, q_2, \cdots, q_M) : \forall i, q_i \in \mathbb{C}, \|q_i\| \leq w\}$.

In the following, we construct an ϵ-net of the metric space $(\mathrm{SP}(\mathcal{Q}), \ell_1)$. Our method will first generate an ϵ-net of $(\text{Raw-}(M, w), \ell_1)$ via a standard procedure and then select those points that are also close to $\mathcal{Q}$-space. We will present and analyze the efficiency of the selection process first and come back to the construction of the ϵ-net afterwards.

Selection process

The selection process determines if some point $\vec{p}$ in Raw-(M, w) is close to $\mathrm{SP}(\mathcal{Q})$. Denote by $\mathrm{dis}(\vec{p})$ the distance of $\vec{p} \in \mathbb{C}^M$ to $\mathrm{SP}(\mathcal{Q})$, i.e.,

$$\mathrm{dis}(\vec{p}) = \min_{\vec{q} \in \mathrm{SP}(\mathcal{Q})} \|\vec{p} - \vec{q}\|_1.$$

We show in this section how to compute $\mathrm{dis}(\vec{p})$ efficiently in space. That the problem admits a polynomial time algorithm follows from the fact that it can be cast as a semidefinite programming

[2] We will abuse the notation later where the metric d is replaced by the norm from which the metric is induced.

problem. However, to the authors' knowledge, only a few restricted classes of SDPs also admit space-efficient algorithms and none of them applies to our case. Thus we need to develop our own space-efficient algorithm for this problem.

By making use of the definition of $\mathrm{SP}(\mathcal{Q})$ and the duality of the ℓ_1 norm, one can find the following equivalent definition of the distance.

$$\mathrm{dis}(\vec{p}) = \min_{\rho \in \mathrm{D}(\mathcal{H})} \max_{\vec{z} \in \mathbf{B}(\mathbb{C}^M, \|\cdot\|_\infty)} \mathrm{Re}\left\langle \vec{p} - \vec{q}(\rho), \vec{z} \right\rangle,$$

where

$$\vec{q}(\rho) = \left(\left\langle Q_1, \rho \right\rangle, \left\langle Q_2, \rho \right\rangle, \cdots, \left\langle Q_M, \rho \right\rangle\right) \in \mathbb{C}^M. \tag{1}$$

By rephrasing $\mathrm{dis}(\vec{p})$ in the above form, one shows the quantity $\mathrm{dis}(\vec{p})$ is actually an equilibrium value. This follows from the well-known extensions of von' Neumann's Min-Max Theorem [vN28, Fan53]. One can easily verify that the density operator set $\mathrm{D}(\mathcal{H})$ and the unit ball of $\mathbb{C}^M$ under ℓ_∞ norm are convex and compact sets. Moreover, the objective function is a bilinear form over the two sets. The Min-Max theorem implies

$$\min_{\rho \in \mathrm{D}(\mathcal{H})} \max_{\vec{z} \in \mathbf{B}(\mathbb{C}^M, \|\cdot\|_\infty)} \mathrm{Re}\left\langle \vec{p} - \vec{q}(\rho), \vec{z} \right\rangle = \max_{\vec{z} \in \mathbf{B}(\mathbb{C}^M, \|\cdot\|_\infty)} \min_{\rho \in \mathrm{D}(\mathcal{H})} \mathrm{Re}\left\langle \vec{p} - \vec{q}(\rho), \vec{z} \right\rangle. \tag{2}$$

Fortunately, there is an efficient algorithm in either time or space (in terms of $d, M, w, 1/\epsilon$) to approximate $\mathrm{dis}(\vec{p})$ with additive error ϵ. The main tool used here is the so-called matrix multiplicative weight update method [AHK05, Kal07, WK06]. Similar min-max forms also appeared before in a series of work on quantum complexity [JW09, Wu10a, Wu10b, GW10]. The algorithm presented here is another simple application of this powerful method. For the sake of completeness, we provide the proof of the following lemma in Appendix A.

Lemma 3. *Given any point $\vec{p} \in$ Raw-(M, w) and $\epsilon > 0$, there is an algorithm (depicted in Appendix A) that approximates $dis(\vec{p})$ with additive error ϵ. Namely, the return value $\tilde{d}$ of this algorithm satisfies*

$$\tilde{d} - \epsilon \leq dis(\vec{p}) \leq \tilde{d} + \epsilon.$$

Moreover, the algorithm runs in $poly(d, M, w, 1/\epsilon)$ time. Furthermore, if d is considered as the input size and $M, w, 1/\epsilon \in O(poly\text{-}log(d))$, this algorithm is also efficient in parallel, namely, it is inside NC.

Construction of the ϵ-net

We are now ready to show the construction of the ϵ-net of $\mathrm{SP}(\mathcal{Q})$. As mentioned before, this construction contains two steps below. Given any $\mathcal{Q}(M, w)$ and $\epsilon > 0$,

- Construct the ϵ-net of the set Raw-(M, w) with the metric induced from the ℓ_1 norm. Denote such an ϵ-net by $\mathcal{R}_\epsilon$.
- For each point $\vec{p} \in \mathcal{R}_\epsilon$, determine $\mathrm{dis}(\vec{p})$ and select it to $\mathcal{N}_\epsilon$ if $\mathrm{dis}(\vec{p}) \leq \epsilon$. We claim $\mathcal{N}_\epsilon$ is the ϵ-net of $(\mathrm{SP}(\mathcal{Q}), \ell_1)$.

The construction for the first step is rather routine. Creating an ϵ'-net T'_ϵ over a bounded complex region $\{z \in \mathbb{C} : \|z\| \leq w\}$ is simple: we can place a 2D grid over the complex plane to cover the disk $\|z\| \leq w$. Simple argument shows $|T'_\epsilon| \in O(\frac{w^2}{\epsilon'^2})$. Then $\mathcal{R}_\epsilon$ can be obtained by the cross-product $\underbrace{T'_\epsilon \times \cdots \times T'_\epsilon}_{M \text{ times}}$. To ensure the closeness in the ℓ_1 norm, we will choose $\epsilon' = \epsilon/M$.

Theorem 4. *The $\mathcal{N}_\epsilon$ constructed above is indeed an ϵ-net of $(\mathrm{SP}(\mathcal{Q}), \ell_1)$ with cardinality at most $O((\frac{w^2M^2}{\epsilon^2})^M)$. Furthermore, for any point $\vec{n} \in \mathcal{N}_\epsilon$, we have $dis(\vec{n}) \leq \epsilon$.*

Proof. First we show $\mathcal{R}_\epsilon$ is indeed an ϵ-net of (Raw-$(M, w), \ell_1$). To that end, consider any point $\vec{p} \in$ Raw-(M, w). From the construction of $\mathcal{R}_\epsilon$, there is some point $\vec{q} \in \mathcal{R}_\epsilon$ such that $\|\vec{p} - \vec{q}\|_\infty \leq \epsilon'$. Then we have $\|\vec{p} - \vec{q}\|_1 \leq M\|\vec{p} - \vec{q}\|_\infty \leq M\epsilon' \leq \epsilon$. Since $\mathcal{N}_\epsilon \subseteq \mathcal{R}_\epsilon$, one has $|\mathcal{N}_\epsilon| \leq |\mathcal{R}_\epsilon| \in O((\frac{w^2M^2}{\epsilon^2})^M)$.

In order to show $\mathcal{N}_\epsilon$ is the required ϵ-net, consider any point $\vec{p} \in \mathrm{SP}(\mathcal{Q})$. Since $\mathrm{SP}(\mathcal{Q}) \subseteq$ Raw-(M, w), there exists a point $\vec{p'} \in \mathcal{R}_\epsilon$ such that $\|\vec{p} - \vec{p'}\|_1 \leq \epsilon$. Hence we have $\mathrm{dis}(\vec{p'}) \leq \epsilon$ and the point $\vec{p'}$ will be selected, namely $\vec{p'} \in \mathcal{N}_\epsilon$. Finally, it is a simple consequence of the selection process that every point $\vec{n} \in \mathcal{N}_\epsilon$ has $\mathrm{dis}(\vec{n}) \leq \epsilon$. □

Remarks. If one choose $\mathcal{Q}$ to be $\mathcal{Q}(d^2, 1) = \{|i\rangle\langle j| : i, j = 1, \cdots, d\}$, one can generate the ϵ-net of the density operator set with the ℓ_1 norm in the method described above. It is akin to generating an ϵ-net for every entry of the density operator. At the other extreme, one can also efficiently generate the ϵ-net of a small size $\mathrm{SP}(\mathcal{Q})$ even when the space dimension d is relatively large.

3 The Main Algorithm

In this section, we prove the main theorem. Without loss of generality, we assume $\mathcal{A}_1, \mathcal{A}_2$ are identical, and of dimension d in Problem 1. Moreover, our algorithm will deal with the set of product states rather than separable states. Namely, we consider the following optimization problem.

$$\begin{aligned} \text{max:} \quad & \langle Q, \rho \rangle \\ \text{subject to:} \quad & \rho = \rho_1 \otimes \rho_2, \rho_1 \in \mathrm{D}(\mathcal{A}_1), \rho_2 \in \mathrm{D}(\mathcal{A}_2). \end{aligned} \tag{3}$$

It is easy to see these two optimization problems are equivalent since product states are extreme points of the set of separable states. Our algorithm works for both maximization and minimization of the objective function. In fact, both results can be obtained at the same time. Since our algorithm naturally extends to multipartite cases, we will demonstrate the algorithm for the k-partite version first, and then obtain the solution for Problem 1 as a special case when $k = 2$.

Problem 2 (k-partite version)**.** Given any Hermitian matrix Q over $\mathcal{A}_1 \otimes \cdots \otimes \mathcal{A}_k$ ($k \geq 2$), compute the optimum value $\mathrm{OptSep}(Q)$ of the following optimization problem to precision δ.

$$\begin{aligned} \text{max:} \quad & \langle Q, \rho \rangle \\ \text{subject to:} \quad & \rho = \rho_1 \otimes \cdots \otimes \rho_k, \forall i, \rho_i \in \mathrm{D}(\mathcal{A}_i). \end{aligned} \tag{4}$$

Before describing the algorithm, we need some terminology about the *decomposability* of a multipartite operator. Any Hermitian operator Q over $\mathcal{A}_1 \otimes \mathcal{A}_2 \otimes \cdots \otimes \mathcal{A}_k$ is called *M-decomposable* if there exists $(Q_1^t, Q_2^t, \cdots, Q_M^t) \in \mathrm{L}(\mathcal{A}_t)^M$ for t=1,2,..., k such that

$$Q = \sum_{i=1}^{M} Q_i^1 \otimes Q_i^2 \otimes \cdots \otimes Q_i^{k-1} \otimes Q_i^k.$$

To facilitate the use of ϵ-net, we adopt a slight variation of the decomposability above. Let $\vec{w} \in \mathbb{R}_+^k$ denote the widths of operators over each $\mathcal{A}_i$. Any Q is called $(M, \vec{w})$*-decomposable* if

1. Let $\mathcal{Q}_t(M, w_t) = (Q_1^t, Q_2^t, \cdots, Q_M^t)$ for t=1,..., k-1. Let $W = \Pi_{i=1}^k w_i$. Generate the ϵ_t-net (by Theorem 4) of $(\mathrm{SP}(\mathcal{Q}_t), \ell_1)$ for each t=1,..., k-1 with $\epsilon_t = w_t\delta/(k-1)W$ and denote such a set by $\mathcal{N}_{\epsilon_t}^t$. Also let OPT store the optimum value of the maximization problem.
2. For each point $\vec{q} = (\vec{q}^1, \vec{q}^2, \cdots \vec{q}^{k-1}) \in \mathcal{N}_{\epsilon_1}^1 \times \mathcal{N}_{\epsilon_2}^2 \times \cdots \times \mathcal{N}_{\epsilon_{k-1}}^{k-1}$, let Q^k be

$$Q^k = \sum_{i=1}^{M} q_i^1 q_i^2 \cdots q_i^{k-1} Q_i^k,$$

   and calculate $\tilde{Q}^k = \frac{1}{2}(Q^k + Q^{k*})$. Then compute the maximum eigenvalue of $\tilde{Q}^k$, denoted by $\lambda_{\max}(\vec{q})$. Update OPT as follows: $\mathrm{OPT} = \max\{\mathrm{OPT}, \lambda_{\max}(\vec{q})\}$.
3. Return OPT.

Figure 1: The main algorithm with precision δ.

Q is M-decomposable and the widths of those operators in the decomposition are bounded in the sense that $\max_i \| Q_i^t \|_{\mathrm{op}} \leq w_t$ for t=1,2,..., k. It is noteworthy to mention that the decomposability defined above is related to the concept tensor rank [3] defined in tensor product spaces. Precisely for any hermitian operator Q over $\mathcal{A}_1 \otimes \mathcal{A}_2 \otimes \cdots \otimes \mathcal{A}_k$, its tensor rank $rank_{\otimes}(Q)$ is defined to be $\min\{M | Q \text{ is } M\text{-decomposable}\}$. Its bounded tensor rank $brank_{\otimes}(Q, \vec{w})$ is defined to be $\min\{M | Q \text{ is } (M, \vec{w})\text{-decomposable}\}$.

By definition, we have $\mathrm{rank}_{\otimes}(Q)$ (resp. $\mathrm{brank}_{\otimes}(Q, \vec{w})$) is the minimum M that Q can be M (resp. $(M, \vec{w})$)-decomposable. However, given the representation Q as input, it is hard in general to compute $\mathrm{rank}_{\otimes}(Q)$, $\mathrm{brank}_{\otimes}(Q, \vec{w})$, or its corresponding decomposition. Therefore it is hard to make use of the optimal decomposition when Q is the only input. Instead, for any $(M, \vec{w})$-decomposable Q we assume its corresponding decomposition is also a part of the input to our algorithm.

Theorem 5. *Let Q be some $(M, \vec{w})$-decomposable Hermitian over $\mathcal{A}_1 \otimes \mathcal{A}_2 \otimes \cdots \otimes \mathcal{A}_k$ (each $\mathcal{A}_i$ is of dimension d) and $\delta > 0$. Also let $(Q_1^t, Q_2^t, \cdots, Q_M^t), t = 1, 2, \cdots, k$ be the operators in the corresponding decomposition of Q. The algorithm shown in Fig. 1 approximates the optimum value $OptSep(Q)$ of Problem 2 with additive error δ. Furthermore, the whole algorithm runs in $O((\frac{(k-1)^2W^2M^2}{\delta^2})^{(k-1)M}) \times poly(d, M, k, W, 1/\delta)$ time.*

Proof. Let's first prove the correctness of the algorithm. By choosing $\mathcal{Q}_t(M, w_t) = (Q_1^t, Q_2^t, \cdots, Q_M^t)$ for t=1,...,k-1, the algorithm first generates the ϵ-net $\mathcal{N}_{\epsilon_t}$ of each $(\mathrm{SP}(\mathcal{Q}_t), \ell_1)$, whose correctness is guaranteed by Theorem 4. By substituting the identity $Q = \sum_{i=1}^M Q_i^1 \otimes Q_i^2 \otimes \cdots \otimes Q_i^{k-1}$, the optimization problem becomes

$$\begin{aligned} \max: \quad & \left\langle \sum_{i=1}^{M} p_i^1 p_i^2 \cdots p_i^{k-1} Q_i^k, \rho_k \right\rangle \\ \text{subject to:} \quad & \forall t \in \{1, \cdots, k-1\}, \vec{p}_t \in \mathrm{SP}(\mathcal{Q}_t(M, w_t)), \text{ and } \rho_k \in \mathrm{D}(\mathcal{A}_k). \end{aligned}$$

[3] Our definition should be more accurately related to the Kronecker-Product rank defined in [RvL11], a special case of the more general concept tensor rank.

Thus, solving the optimization problem amounts to first enumerating $\vec{p}_t \in \mathrm{SP}(\mathcal{Q}_t(M, w_1))$ for each t, and then solving the optimization problem over $\mathrm{D}(\mathcal{A}_k)$.

Consider any point $\vec{p} = (\vec{p}^1, \vec{p}^2, \cdots, \vec{p}^{k-1}) \in \mathrm{SP}(\mathcal{Q}_1) \times \cdots \times \mathrm{SP}(\mathcal{Q}_{k-1})$. Due to Theorem 4, there is at least one point $\vec{q} = (\vec{q}^1, \vec{q}^2, \cdots \vec{q}^{k-1}) \in \mathcal{N}^1_{\epsilon_1} \times \mathcal{N}^2_{\epsilon_2} \times \cdots \times \mathcal{N}^{k-1}_{\epsilon_{k-1}}$ such that $\|\vec{q}^t - \vec{p}^t\|_1 \leq \epsilon_t$ for t=1,..,k-1. The choice of $\tilde{Q}^k$ is to symmetrize Q^k where the latter is not guaranteed to be Hermitian because $\vec{q}$ only comes from an ϵ-net. With $\tilde{Q}^k$ being Hermitian, it is clear that $\lambda_{\max}(\vec{q}) = \max_{\rho_k \in \mathrm{D}(\mathcal{A}_k)} \langle \tilde{Q}^k, \rho_k \rangle$. Now let's analyze how much error will be induced in this process.

Let $P^k(\vec{p}) = \sum_{i=1}^M p_i^1 p_i^2 \cdots p_i^{k-1} Q_i^k$ and $\tilde{P}^k = \frac{1}{2}(P^k + P^{k*})$. It is not hard to see that $P^k = \tilde{P}^k$. The error bound is achieved by applying a chain of triangle inequalities as follows. Firstly, one has

$$\|\tilde{P}^k - \tilde{Q}^k\|_{\mathrm{op}} = \|\frac{1}{2}(P^k - Q^k) + \frac{1}{2}(P^{k*} - Q^{k*})\|_{\mathrm{op}} \leq \frac{1}{2}(\|P^k - Q^k\|_{\mathrm{op}} + \|P^{k*} - Q^{k*}\|_{\mathrm{op}}) = \|P^k - Q^k\|_{\mathrm{op}}.$$

Then we substitute the expressions for P^k, Q^k and apply the standard hybrid argument.

$$\begin{aligned} \|P^k - Q^k\|_{\mathrm{op}} &= \|\sum_{i=1}^M (p_i^1 p_i^2 \cdots p_i^{k-1} - q_i^1 q_i^2 \cdots q_i^{k-1}) Q_i^k\|_{\mathrm{op}} \\ &= \|\sum_{i=1}^M \sum_{t=1}^{k-1} (q_i^1 \cdots q_i^{t-1} p_i^t p_i^{t+1} \cdots p_i^{k-1} - q_i^1 \cdots q_i^{t-1} q_i^t p_i^{t+1} \cdots p_i^{k-1}) Q_i^k\|_{\mathrm{op}}, \end{aligned}$$

which is immediately upper bounded by the sum of the following terms,

$$\sum_{i=1}^M |p_i^1 - q_i^1||p_i^2 \cdots p_i^{k-1}|\|Q_i^k\|_{\mathrm{op}}, \sum_{i=1}^M |q_i^1||p_i^2 - q_i^2||p_i^3 \cdots p_i^{k-1}|\|Q_i^k\|_{\mathrm{op}}, \cdots, \sum_{i=1}^M |q_i^1 \cdots q_i^{k-2}||p_i^{k-1} - q_i^{k-1}|\|Q_i^k\|_{\mathrm{op}}.$$

As the t^{th} term above can be upper bounded by $\epsilon_t W / w_t$ for each t=1,...,k-1, we have,

$$\|\tilde{P}^k - \tilde{Q}^k\|_{\mathrm{op}} \leq \epsilon_1 W/w_1 + \epsilon_2 W/w_2 + \cdots + \epsilon_{k-1} W/w_{k-1} = \underbrace{\frac{\delta}{k-1} + \cdots + \frac{\delta}{k-1}}_{\text{k-1 terms}} = \delta.$$

Hence the optimum value for any fixed $\vec{p}$ won't differ too much from the one for its approximation $\vec{q}$ in the ϵ-net. This is because

$$\max_{\rho_k \in \mathrm{D}(\mathcal{A}_k)} \left\langle \tilde{P}^k, \rho_k \right\rangle = \max_{\rho_k \in \mathrm{D}(\mathcal{A}_k)} \left\langle \tilde{Q}^k, \rho_k \right\rangle + \left\langle \tilde{P}^k - \tilde{Q}^k, \rho_k \right\rangle.$$

By Hölder Inequalities we have $|\langle \tilde{P}^k - \tilde{Q}^k, \rho_k \rangle| \leq \|\tilde{P}^k - \tilde{Q}^k\|_{\mathrm{op}} \|\rho_k\|_{\mathrm{tr}} \leq \delta$ and thus,

$$\lambda_{\max}(\vec{q}) - \delta \leq \max_{\rho_k \in \mathrm{D}(\mathcal{A}_k)} \left\langle \tilde{P}^k(\vec{p}), \rho_k \right\rangle \leq \lambda_{\max}(\vec{q}) + \delta.$$

We now optimize $\vec{p}$ over $\mathrm{SP}(\mathcal{Q}_1) \times \cdots \times \mathrm{SP}(\mathcal{Q}_{k-1})$ and the corresponding $\vec{q}$ will run over the ϵ-net $\mathcal{N}^1_{\epsilon_1} \times \mathcal{N}^2_{\epsilon_2} \times \cdots \times \mathcal{N}^{k-1}_{\epsilon_{k-1}}$. As every point $\vec{q} \in \mathcal{N}^1_{\epsilon_1} \times \mathcal{N}^2_{\epsilon_2} \times \cdots \times \mathcal{N}^{k-1}_{\epsilon_{k-1}}$ is also close to $\mathrm{SP}(\mathcal{Q}_1) \times \cdots \times \mathrm{SP}(\mathcal{Q}_{k-1})$ in the sense that $\mathrm{dis}(\vec{q}^t) \leq \epsilon_t$ for each t, we have

$$\max_{\vec{q} \in \mathcal{N}^1_{\epsilon_1} \times \mathcal{N}^2_{\epsilon_2} \times \cdots \times \mathcal{N}^{k-1}_{\epsilon_{k-1}}} \lambda_{\max}(\vec{q}) - \delta \leq \max_{\vec{p} \in \mathrm{SP}(\mathcal{Q}_1) \times \cdots \times \mathrm{SP}(\mathcal{Q}_{k-1})} \max_{\rho_k \in \mathrm{D}(\mathcal{A}_k)} \left\langle \tilde{P}^k(\vec{p}), \rho_k \right\rangle \leq \max_{\vec{q} \in \mathcal{N}^1_{\epsilon_1} \times \mathcal{N}^2_{\epsilon_2} \times \cdots \times \mathcal{N}^{k-1}_{\epsilon_{k-1}}} \lambda_{\max}(\vec{q}) + \delta.$$

Finally, it is not hard to see that $\mathrm{OPT} = \max_{\vec{q} \in \mathcal{N}^1_{\epsilon_1} \times \mathcal{N}^2_{\epsilon_2} \times \cdots \times \mathcal{N}^{k-1}_{\epsilon_{k-1}}} \lambda_{\max}(\vec{q})$ and therefore

$$\mathrm{OPT} - \delta \leq \mathrm{OptSep}(Q) \leq \mathrm{OPT} + \delta.$$

Now let us analyze the efficiency of this algorithm. The total number of points in the ϵ-net $\mathcal{N}^1_{\epsilon_1} \times \mathcal{N}^2_{\epsilon_2} \times \cdots \times \mathcal{N}^{k-1}_{\epsilon_{k-1}}$ is upper bounded by $O((\frac{(k-1)^2W^2M^2}{\delta^2})^{(k-1)M})$ by Theorem 4. For each point $\vec{q}$, the generation of such a point will cost time polynomial in $d, M, W, 1/\delta$ (this part is done through the calculation of $\mathrm{dis}(\vec{q})$. See Lemma 3.). After the generation process, one needs to calculate $\tilde{Q}^k$ and its maximum eigenvalue for each point, which can be done in time polynomial in d, k, M. Thus, the total running time is bounded by $O((\frac{(k-1)^2W^2M^2}{\delta^2})^{(k-1)M}) \times \mathrm{poly}(d, M, k, W, 1/\delta)$. □

Remarks. There are a few remarks to make about Theorem 5. First, it is straightforward to extend the concept of decomposability to its approximate version. For instance, any Hermitian Q is called *ϵ-approximate* $(M, \vec{w})$-decomposable if there exists some $(M, \vec{w})$-decomposable $\tilde{Q}$, such that $\|Q - \tilde{Q}\| \leq \epsilon$, where the norm could be either the operator norm or the *injective* tensor norm. It is easy to verify that the same algorithm solves $\mathrm{OptSep}(Q)$ approximately.

Second, all operations in the algorithm described in Fig. 1 can be implemented efficiently in parallel in some situation. This is because fundamental operations of matrices can be done in NC and the calculation of $\mathrm{dis}(\vec{p})$ can be done in NC (See Lemma 3) when $M, W, k, 1/\delta$ are in nice forms of d. Thus, we can apply the observation stated in the introduction and prove the algorithm in Fig. 1 can also be made space-efficient. To facilitate the later use of this result, we will change the input size as follows.

Corollary 4. Let n be the input size such that $d = \exp(\mathrm{poly(n)})$, if $W/\delta \in O(\mathrm{poly}(n)), kM \in O(\mathrm{poly(n)})$, then $\mathrm{OptSep}(Q)$ can be approximated with additive error δ in PSPACE.

Proof. Here we present an argument that composes space-efficient algorithms directly. Given Q and its decomposition as input, consider the following algorithm

1. Enumerate each point $\vec{p} = (\vec{p}_1, \vec{p}_2, \cdots, \vec{p}_{k-1})$ from the raw set $\mathcal{R}^1_{\epsilon_1} \times \cdots \times \mathcal{R}^{k-1}_{\epsilon_{k-1}}$.

2. Compute $\mathrm{dis}(\vec{p}_t)$ for each t=1,...,k-1. If $\vec{p}$ is a valid point in the epsilon-net, then we continue with the rest part in Step 2 of the algorithm in Fig. 1.

3. Compare the values obtained by each point $\vec{p}$ and keep the optimum one.

Given the condition $W/\delta \in O(\mathrm{poly}(n)), kM \in O(\mathrm{poly(n)})$, the first part of the algorithm can be done in polynomial space. This is because in this case each point in the raw set can be represented by polynomial space and therefore enumerated in polynomial space. The second part is more difficult. Computing $\mathrm{dis}(\vec{p}_t)$ for each t=1,...,k-1 can be done in NC(poly(n)) as shown in Lemma 3. Step 2 in the main algorithm only contains fundamental operations of matrices and the spectrum decomposition. Thus, it also admits a parallel algorithm in NC(poly(n)). One can easily compose the two circuits and get a polynomial space implementation by the relation NC(poly)=PSPACE [Bor77]. The third part can obviously be done in polynomial space.

Thus, by composing these three polynomial-space implementable parts, one proves the whole algorithm can be done in PSPACE. □

4 Simulation of several variants of QMA(2)

This section illustrates how one can make use of the algorithm shown in Section 3 (when k=2) to simulate some variants of the complexity class QMA(2). The idea is to show for those variants, the corresponding POVM matrices of acceptance are $(M, \vec{w})$-decomposable with small Ms. Before we dive into the details, let us recall the definition of the complexity class QMA(2).

Definition 2. A language $\mathcal{L}$ is in $\mathrm{QMA}(2)_{m,c,s}$ if there exists a polynomial-time generated family of quantum verification circuits $Q = \{Q_n | n \in \mathbb{N}\}$ such that for any input x of size n, the circuit Q_n implements a two-outcome measurement $\{Q_x^{\mathrm{acc}}, \mathbb{1} - Q_x^{\mathrm{acc}}\}$. Furthermore,

- Completeness: If $x \in \mathcal{L}$, there exist 2 witness $|\psi_1\rangle \in \mathcal{A}_1, |\psi_2\rangle \in \mathcal{A}_2$, each of m qubits, such that
$$\langle Q_x^{\mathrm{acc}}, |\psi_1\rangle\langle\psi_1| \otimes |\psi_2\rangle\langle\psi_2|\rangle \geq c.$$
- Soundness: If $x \notin \mathcal{L}$, then for any states $|\psi_1\rangle \in \mathcal{A}_1, |\psi_2\rangle \in \mathcal{A}_2$,
$$\langle Q_x^{\mathrm{acc}}, |\psi_1\rangle\langle\psi_1| \otimes |\psi_2\rangle\langle\psi_2|\rangle \leq s.$$

We call QMA(2)=$\mathrm{QMA}(2)_{\mathrm{poly}(n),2/3,1/3}$. It is easy to see that simulating the complexity class QMA(2) amounts to distinguishing between the two promises of the maximum acceptance probability, represented by the inner product $\langle Q_x^{\mathrm{acc}}, \rho\rangle$, over the set of all possible valid strategies of the two provers, which is exactly $\mathrm{SepD}(\mathcal{A}_1 \otimes \mathcal{A}_2)$. Note the maximum acceptance probability is exactly $\mathrm{OptSep}(Q_x^{\mathrm{acc}})$ defined in Problem 1. Thus, if one were able to distinguish between the two promises of $\mathrm{OptSep}(Q_x^{\mathrm{acc}})$, one could simulate this protocol with the same amount of resources (time or space).

The first example is the variant with only logarithm-size proofs, namely $\mathrm{QMA}(2)_{O(\log(n)),2/3,1/3}$. It is not hard to find out the corresponding POVMs of acceptance (i.e. Q_x^{acc}) need to be (poly(n),$\vec{w}$)-decomposable since $\mathcal{A}_1, \mathcal{A}_2$ in this case are only of polynomial dimension. Moreover, $\vec{w}$ could be $(1, 1)$ in this case. Thus, it follows directly from Corollary 4 that $\mathrm{OptSep}(Q_x^{\mathrm{acc}})$ can be approximated in polynomial space. Namely,

$$\mathrm{QMA}(2)_{O(\log(n)),2/3,1/3} \subseteq \mathrm{PSPACE}.$$

The next example is slightly less trivial. Before moving on, we need some terminology about the quantum verification circuits Q. Assume the input x is fixed from now on. Let $\mathcal{A}_1, \mathcal{A}_2$ be the Hilbert space of size $d_{\mathcal{A}}$ for the two proofs and let $\mathcal{V}$ be the ancillary space of size $d_{\mathcal{V}}$. Note $d_{\mathcal{A}} d_{\mathcal{V}}$ is exponential in n. Then the quantum verification process will be carried out on the space $\mathcal{A}_1 \otimes \mathcal{A}_2 \otimes \mathcal{V}$ with some initial state $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \left|\vec{0}\right\rangle$ where $|\psi_1\rangle, |\psi_2\rangle$ are provided by the provers. The verification process is also efficient in the sense that the whole circuit only consists of polynomial elementary gates. Without loss of generality, we can fix one universal gate set for the verification circuits in advance. Particularly, we choose the universal gate set to be single qubit gates plus the CNOT gates [NC00]. One can also choose other universal gate sets without any change of the main result.

We categorize all elementary gates in the verification circuits into two types. A gate is of *type-I* if it only affects the qubits within the same space (i.e, $\mathcal{A}_1, \mathcal{A}_2$, or, $\mathcal{V}$). Otherwise, this gate is of *type-II*. It is easy to see single qubit gates are always type-I gates. The only type-II gates are CNOT

gates whose control qubit and target qubit sit in different spaces. Let $p, r : \mathbb{N} \to \mathbb{N}$ be polynomial-bounded functions. A polynomial-time generated family of quantum verification circuits Q is called $Q[p, r]$ if each Q_n only contains $p(n)$ type-I elementary gates and $r(n)$ type-II elementary gates.

Definition 3. A language $\mathcal{L}$ is in $\mathrm{QMA}(2)_{m,c,s}[p, r]$ if $\mathcal{L}$ is in $\mathrm{QMA}(2)_{m,c,s}$ with some $Q[p, r]$ verification circuit family.

It is easy to see that QMA(2) = QMA(2)[poly, poly] from our definition. In the following we will show that when the number of type-II gates is relatively small, one can simulate this complexity model efficiently by the algorithm in Fig. 1.

Lemma 6. *For any family of verification circuits $Q[p, r]$, the corresponding POVM Q_x^{acc} is $(4^{r(n)}, (1, 1))$-decomposable for any $n \in \mathbb{N}$ and input x. Moreover, this decomposition can be calculated in parallel with $O(t(n)4^{r(n)}) \times poly(n)$ time.*

Proof. For any $n \in \mathbb{N}$ and input x, let us denote the whole unitary that the verification circuit applies on the initial state by $U = U_t U_{t-1} \cdots U_1$ where each U_i corresponds to one elementary gate and $t = p + r$. Without loss of generality, we assume the output bit is the first qubit in the space $\mathcal{V}$ and the verification accepts when that qubit is 1. Let $\bar{\mathcal{V}}$ be the space $\mathcal{V}$ without the first qubit, then we have

$$Q_x^{\mathrm{acc}} = \mathrm{Tr}_{\mathcal{V}} \left(\mathbb{1}_{\mathcal{A}_1\mathcal{A}_2} \otimes \left|\vec{0}\right\rangle\left\langle\vec{0}\right| \left(U^* \mathbb{1}_{\mathcal{A}_1\mathcal{A}_2} \otimes \mathbb{1}_{\bar{\mathcal{V}}} \otimes |1\rangle\langle 1| \, U\right) \mathbb{1}_{\mathcal{A}_1\mathcal{A}_2} \otimes \left|\vec{0}\right\rangle\left\langle\vec{0}\right| \right).$$

Let $P_{t+1} = \mathbb{1}_{\mathcal{A}_1\mathcal{A}_2} \otimes \mathbb{1}_{\bar{\mathcal{V}}} \otimes |1\rangle\langle 1|$ and $P_\tau = U_\tau^* P_{\tau+1} U_\tau$ for τ=t,t-1,...,1. It is easy to see $P_1 = U^*(\mathbb{1}_{\mathcal{A}_1\mathcal{A}_2} \otimes \mathbb{1}_{\bar{\mathcal{V}}} \otimes |1\rangle\langle 1|)U$. Also it is straightforward to verify that P_{t+1} is 1-decomposable. Now let us observe how the decomposability of P_τ changes with τ.

For each τ, the unitary U_τ either corresponds to a type-I or type-II elementary gate. In the former case, applying U_τ won't change the decomposability. Thus, P_τ is M-decomposable if $P_{\tau+1}$ is. In the latter case, applying U_τ will potentially change the decomposability in the following sense. For any such CNOT gate one has $U_\tau = |0\rangle\langle 0| \otimes \mathbb{1} + |1\rangle\langle 1| \otimes X$ where X is the Pauli matrix for the flip. And one can show

$$\begin{aligned} P_\tau &= (|0\rangle\langle 0| \otimes \mathbb{1}) P_{\tau+1} (|0\rangle\langle 0| \otimes \mathbb{1}) + (|0\rangle\langle 0| \otimes \mathbb{1}) P_{\tau+1} (|1\rangle\langle 1| \otimes X) \\ &+ (|1\rangle\langle 1| \otimes X) P_{\tau+1} (|0\rangle\langle 0| \otimes \mathbb{1}) + (|1\rangle\langle 1| \otimes X) P_{\tau+1} (|1\rangle\langle 1| \otimes X). \end{aligned}$$

Thus in general we can only say P_τ is $4M$-decomposable if $P_{\tau+1}$ is M-decomposable. As there are $r(n)$ type-II gates, one immediately has P_1 is $4^{r(n)}$-decomposable. Moreover, each operator appearing in the decomposition is a multiplication of unitaries , $|0\rangle\langle 0|$, $|1\rangle\langle 1|$ and X in some order, which implies the operator norm of those operators is bounded by 1. Therefore we have P_1 is $(4^{r(n)}, (1, 1))$-decomposable.

Finally, it is not hard to verify that multiplications with $\mathbb{1}_{\mathcal{A}_1\mathcal{A}_2} \otimes \left|\vec{0}\right\rangle\left\langle\vec{0}\right|$ and partial trace over $\mathcal{V}$ won't change the decomposability of P_1. Namely, we have Q_x^{acc} is $(4^{r(n)}, (1, 1))$-decomposable. The above proof can also be considered as the process to compute the decomposition of Q_x^{acc}. Each multiplication of matrices can be done in NC(poly(n)). And the total number of multiplications is upper bounded by $O(t(n)4^{r(n)})$. Therefore, the total parallel running time is upper bounded by $O(t(n)4^{r(n)}) \times$ poly(n). □

Corollary 5. QMA(2)$[\text{poly}(n), O(\log(n))] \subseteq$ PSPACE.

Proof. This is a simple consequence of Lemma 7 and Corollary 4. For any fixed x of length n. One can first compute the decomposition of Q_x^{acc} in parallel with $O(t(n)4^{r(n)})\times$ poly(n) time, which is parallel polynomial time in n when $r(n) = O(\log(n))$ and $t(n) \in \text{poly}(n)$. Hence the first step can be done in polynomial space via the relation NC(poly)=PSPACE [Bor77].

Then one can invoke the parallel algorithm in Corollary 4 to approximate OptSep(Q_x^{acc}) to sufficient precision δ such that one can distinguish between the two promises. Precisely in this case, we choose those parameters as follows,

$$k = 2, W = 1, M = 4^{O(\log(n))} = \text{poly}(n), 1/\delta = \text{poly}(n).$$

Thus the whole algorithm can be done in polynomial space, which completes the proof. □

Remarks. Although the proof of the result is not too technical, it establishes the first non-trivial upper bound (PSPACE in this case) for variants of QMA(2) that allow quantum operations acting on both proofs at the same time. In contrast, previous results are all about variants with nonadaptive or adaptive local measurements, like BellQMA(2) [Bra08, ABD+09, CD10] or LOCC-QMA(2) [ABD+09, BCY11].

However, our results are hard to extend to the most general case of QMA(2). This is because SWAP-test operation uses many more type-II gates than what is allowed in our method. And SWAP-test seems to be inevitable if one wants to fully characterize the power of QMA(2).

5 Quasi-polynomial algorithms for local Hamiltonian cases

In this section, we illustrate that if Q appears in the objective function that is a local Hamiltonian then the optimal value OptSep(Q) can be efficiently computed by our main algorithm. Consider any k-partite space $\mathcal{A}_1 \otimes \mathcal{A}_2 \otimes \cdots \otimes \mathcal{A}_k$ where each partite $\mathcal{A}_i$ contains n qubits and thus is of dimension 2^n.

Definition 4. Any Hermitian Q over $\mathcal{A}_1 \otimes \cdots \otimes \mathcal{A}_k$ is a l-local Hamiltonian if Q is expressible as $Q = \sum_{i=1}^{r} H_i$ where each term is a Hermitian operator acting on at most l qubits among k parties.

Hamiltonians are widely studied in physics since they usually characterize the energy of a physical system. Local Hamiltonians are of particular interest since they refer to the energy of many interesting models in low-dimension systems. Our algorithm can be considered as a way to find the minimum energy in the system achieved by separable states.

Local Hamiltonians are also appealing to computational complexity theorists since the discovery of the promise 5-local Hamiltonian problem [KSV02] which turns out to be QMA-complete. Precisely, it refers to the following promise problem when $k = 1, l = 5$.

Problem 3 (k-partite l-local Hamiltonian problem)**.** Take the expression $Q = \sum_{i=1}^{r} H_i$ for any l-local Hamiltonian over $\mathcal{A}_1 \otimes \cdots \otimes \mathcal{A}_k$ as input[4], where $\|H_i\|_{\text{op}} \leq 1$ for each i. Let OptSep(Q) denote the minimum value of $\langle Q, \rho \rangle$ achieved for some $\rho \in \text{SepD}(\mathcal{A}_1 \otimes \cdots \otimes \mathcal{A}_k)$. The goal is to tell between the following two promises: either OptSep$(Q) \geq a$ or OptSep$(Q) \leq b$ for some $a > b$ with inverse polynomial gap.

[4]It is noteworthy to mention that the input size of local Hamiltonian problems can be only poly-logarithm in the dimension of the space where Q sits in.

When $k = 1$, the promise problem defined above is exactly the original l-local Hamiltonian problem. Subsequent results demonstrate that it remains QMA-complete even when $l = 3, 2$ [AGIK09, KKR06, OT08]. Our definition of the promise problem naturally extends to the k-partite case. We refer to Chapter 14 in [KSV02] for technical details. It is not hard to see that k-partite l-local Hamiltonian problems belong to QMA(k) by applying similar techniques in the original proof. However, they do not remain as QMA(k)-complete problems. This is because the original reduction transforms from the proof space to the transcript and clock space and the separability of quantum states does not persevere under such an operation. As a result, k-partite local Hamiltonian problems defined above only enforce the separability in the transcript and clock space rather than in the proof space. Note this is not an issue for the 1-partite case since there is no separability involved. Nevertheless it becomes a huge problem for its k-partite extensions.

Lemma 7. *Any l-local Hamiltonian Q over $\mathcal{A}_1 \otimes \cdots \otimes \mathcal{A}_k$ such that $Q = \sum_{i=1}^{r} H_i$ and $\|H_i\|_{op} \leq w$ is $(O((4nk)^l), w)$-decomposable.*

Proof. Since Q is a l-local Hamiltonian, it is easy to see $r \leq \binom{kn}{l}$. For each H_i with $\|H_i\|_{\text{op}} \leq w$, since it acts only on at most l qubits, it must be $(4^l, w)$-decomposable. Thus Q is $(r4^l, w)$-decomposable. In terms of only n, k, l, we have Q is $(O((4nk)^l), w)$-decomposable. □

Corollary 6. Take the expression $Q = \sum_{i=1}^{r} H_i$ of any l-local Hamiltonian over $\mathcal{A}_1 \otimes \cdots \otimes \mathcal{A}_k$ (each $\mathcal{A}_i$ is of dimension $d = 2^n$) such that $\|H_i\|_{\text{op}} \leq w$ for each i as input. Assuming $k, l = O(1)$, the quantity $\text{OptSep}(Q)$ can be approximated to precision δ in quasi-polynomial time in $d, w, 1/\delta$.

If n is considered as the input size and $w/\delta = O(\text{poly(n)})$, then $\text{OptSep}(Q)$ can be approximated to precision δ in PSPACE.

Proof. The proof of the first part follows directly from Lemma 7 and Theorem 5. Recall the proof of Lemma 7 also provides a way to compute the decomposition of Q given the expression $Q = \sum_{i=1}^{r} H_i$ as input. It is easy to verify that $O(r4^l)$ time (upper bounded by $O((4k \log d)^l)$) is sufficient to complete this computation. After that, one may directly invoke the algorithm in Fig. 1 and make use of Theorem 5. Now we substitute the following identities into our main algorithm. Note $k, l = O(1)$ and we have $M = O(\log^{O(1)} d), W = w^{O(1)}$. One immediately gets the total running time bounded by

$$\exp(O(\log^{O(1)}(d)(\log\log d + \log w/\delta))) \times \text{poly}(d, w, 1/\delta),$$

which is quasi-polynomial time in $d, w, 1/\delta$.

For the second part when n is considered as the input size, it is easy to see the computation of the decomposition of Q according to Lemma 7 can be done in NC(poly), henceforth in polynomial space. (Note $M = O(\text{poly(n)})$.) Then by composing with the polynomial-space algorithm implied by Corollary 4, one proves the whole algorithm can be implemented in polynomial space. □

Remarks. It is a direct consequence of Corollary 6 that Problem 3 is inside PSPACE.

6 An algorithm with running time exponential in $\|Q\|_{\mathbf{F}}$

In this section we demonstrate another application of the simple idea "enumeration" by epsilon-net to Problem 1. As a result, we obtained an algorithm with running time exponential in $\|Q\|_{\text{F}}$

1. Compute the spectral decomposition of Q. After that, one has the decomposition $Q = \sum_t \lambda_t |\Psi_t\rangle\langle\Psi_t|$. Choose $\epsilon = \delta/2$ and $\Gamma_\epsilon = \{t : \lambda_t \geq \epsilon\}$.Also let OPT store the optimum value of the maximization problem.
2. Generate the ε-net of the unit ball of $\mathbb{C}^{|\Gamma_\epsilon|}$ under the Euclidean norm with $\varepsilon = \frac{\delta}{4\|Q\|_F}$. Denote such set by $\mathcal{N}_\varepsilon$. Then for each point $\alpha \in \mathcal{N}_\varepsilon$,
   (a) Compute $|\phi_\alpha\rangle = \sum_{t\in\Gamma_\epsilon} \alpha_t^* \sqrt{\lambda_t} |\Psi_t\rangle$ and compute the Schmidt decomposition of $|\phi_\alpha\rangle$, i.e.
   $$|\psi_\alpha\rangle = \sum_i \mu_i |u_i\rangle |v_i\rangle ,$$
   where $\mu_1 \geq \mu_2 \geq \cdots$ and $\{u_i\}, \{v_i\}$ are orthogonal bases. Note $|\phi_\alpha\rangle$ is not necessarily a unit vector.
   (b) Update OPT as follows: OPT=max{OPT,μ_1}.
3. Return OPT.

Figure 2: The algorithm runs in time exponential in $\|Q\|_F/\delta$.

(or $\|Q\|_{\text{LOCC}}$ [MWW09][5]) for computing OptSep(Q) with additive error δ. A similar running time $\exp(O(\log^2(d)\delta^{-2}\|Q\|_F^2))$ was obtained in [BCY11] using some known results in quantum information theory.(i.e., the semidefinite programming for finding symmetric extension [DPS04] and an improved quantum de Finetti-type bound.)

By contrast, our algorithm makes no use of any advanced tool above and only utilizes fundamental operations of matrices. Intuitively, in order to approximate the optimum value to precision δ, one only needs to look at the eigenspace of eigenvalues greater than δ, the dimension of which is no more than $\|Q\|_F^2/\delta^2$. Nevertheless, naively enumerating density operators over that subspace doesn't work since one cannot detect the separability of those density operators. We circumvent this difficulty by making nontrivial use of the Schmidt decomposition of bipartite pure states.

Finally, as mentioned in the introduction we admit that other results in the original paper [BCY11] do not follow from our algorithm and our method cannot be seen as a replacement of the kernel technique of that paper. Also our method does not extend to the k-partite version as there is no Schmidt decomposition in that case.

Recall the optimization problem we are interested in is equivalent to the following one.

$$\max : \langle Q, \rho\rangle \text{ s.t. } \rho = |u\rangle\langle u| \otimes |v\rangle\langle v| , |u\rangle \in \mathcal{A}_1, |v\rangle \in \mathcal{A}_2.$$

Theorem 8. *Given any positive semidefinite Q over $\mathcal{A}_1 \otimes \mathcal{A}_2$ (of dimension $d \times d$) and $\delta > 0$, the algorithm in Fig. 2 approximates the optimal value OptSep(Q) with additive error δ with running time* $\exp(O(\log(d) + \delta^{-2}\|Q\|_F^2 \ln(\|Q\|_F/\delta)))$.

Proof. We first prove the correctness of the algorithm. The analysis will mainly be divided into two parts. Let $S_\epsilon = \text{span}\{|\Psi_t\rangle : t \in \Gamma_\epsilon\}$. The first part shows it suffices to only consider vectors inside

[5]This follows easily from the fact $\|Q\|_F = O(\|Q\|_{\text{LOCC}})$ [MWW09] where $\|Q\|_{\text{LOCC}}$ stands for the LOCC norm of the operator Q.

the subspace S_ϵ for approximating $\mathrm{OptSep}(Q)$ with additive error δ. The second one demonstrates that our algorithm in Fig. 2 approximates the optimal value obtained by only considering vectors in S_ϵ. Precisely, since $\{|\Psi_i\rangle\}$ forms a basis, one has $|u\rangle|v\rangle = \sum_{t\in\Gamma_\epsilon}\beta_t|\Psi_t\rangle + \sum_{t\notin\Gamma_\epsilon}\beta_t|\Psi_t\rangle$ where β is a unit vector in $\mathbb{C}^{d^2}$. Then we have

$$\langle Q, |u\rangle\langle u|\otimes|v\rangle\langle v|\rangle = \underbrace{\sum_{t\in\Gamma_\epsilon}\lambda_t|\beta_t|^2}_{(I)} + \underbrace{\sum_{t\notin\Gamma_\epsilon}\lambda_t|\beta_t|^2}_{(II)},$$

where the term (II) is obviously bounded by $\delta/2$ (i.e., $\sum_{t\notin\Gamma_\epsilon}\lambda_t|\beta_t|^2 \le \delta/2$). For the term (I), it is equivalent to $\mathrm{OptSep}(\tilde{Q})$ where $\tilde{Q} = \sum_{t\in\Gamma_\epsilon}\lambda_t|\Psi_t\rangle\langle\Psi_t|$. Namely, small eigenvalues are truncated in $\tilde{Q}$. Now observe the following identity.

$$\begin{aligned}
\max_{|u\rangle|v\rangle}\langle\tilde{Q}, |u\rangle\langle u|\otimes|v\rangle\langle v|\rangle &= \max_{|u\rangle|v\rangle}\sum_{t\in\Gamma_\epsilon}\lambda_t|\langle u|\langle v|\Psi_t\rangle|^2 = \max_{|u\rangle|v\rangle}\|\gamma^{u,v}\|^2 \\
&= \max_{|u\rangle|v\rangle}\max_{\alpha\in\mathbf{B}(\mathbb{C}^{|\Gamma_\epsilon|},\|\cdot\|)}|\sum_{t\in\Gamma_\epsilon}\alpha_t^*\sqrt{\lambda_t}\langle u|\langle v|\Psi_t\rangle|^2 = \max_{|u\rangle|v\rangle}\max_{\alpha\in\mathbf{B}(\mathbb{C}^{|\Gamma_\epsilon|},\|\cdot\|)}|\langle u|\langle v|\phi_\alpha\rangle|^2 \\
&= \max_{\alpha\in\mathbf{B}(\mathbb{C}^{|\Gamma_\epsilon|},\|\cdot\|)}\max_{|u\rangle|v\rangle}|\langle u|\langle v|\phi_\alpha\rangle|^2,
\end{aligned}$$

where $\gamma^{u,v}\in\mathbb{C}^{|\Gamma_\epsilon|}$ and $\gamma_t^{u,v} = \sqrt{\lambda_t}\langle u|\langle v|\Psi_t\rangle$ for each $t\in\Gamma_\epsilon$. The second line comes from the duality of the Euclidean norm (i.e., $\|y\| = \max_{\|z\|\le 1}|\langle z|y\rangle|$). The third line comes by exchanging positions of the two maximizations. We then make use of the following well-known fact.

Fact ([NC00])**.** For any bipartite vector $|\psi\rangle$ with the Schmidt decomposition

$$|\psi\rangle = \sum_i \mu_i|u_i\rangle|v_i\rangle,$$

where $\mu_1\ge\mu_2\ge\cdots$ and $\{u_i\},\{v_i\}$ are orthogonal bases . Then $\max_{|u\rangle|v\rangle}|\langle u|\langle v|\psi\rangle| = \mu_1$ and the maximum value is obtained by choosing $|u\rangle|v\rangle$ to be $|u_1\rangle|v_1\rangle$.

It is not hard to see that our algorithm computes exactly the term on the third line except that we replace the unit ball by its ϵ-net. However, this won't incur too much extra error. For any $\alpha\in\mathbf{B}(\mathbb{C}^{|\Gamma_\epsilon|},\|\cdot\|)$, there exists $\tilde{\alpha}\in\mathcal{N}_\epsilon$, such that $\|\alpha-\tilde{\alpha}\|\le\varepsilon$. Thus, the extra error incurred is $||\langle u|\langle v|\phi_\alpha\rangle|^2 - |\langle u|\langle v|\phi_{\tilde{\alpha}}\rangle|^2|$ and can be bounded by

$$\begin{aligned}
(\||\phi_\alpha\rangle\| + \||\phi_{\tilde{\alpha}}\rangle\|)|\langle u|\langle v|\psi_\alpha - \psi_{\tilde{\alpha}}\rangle| &\le 2\max_{\|\beta_1\|\le 1}\|\phi_{\beta_1}\|\max_{\beta_2=\alpha-\tilde{\alpha},\|\beta_2\|\le\varepsilon}\|\phi_{\beta_2}\| \\
&= 2\sqrt{\|Q\|_{\mathrm{F}}}\times\varepsilon\sqrt{\|Q\|_{\mathrm{F}}} \le \delta/2,
\end{aligned}$$

where $\max_{\|\beta\|\le\epsilon'}\|\phi_\beta\|\le\epsilon'\sqrt{\|Q\|_{\mathrm{F}}}$ for any $\epsilon'>0$ can be verified directly and therefore the total additive error is bounded by $\delta/2+\delta/2=\delta$.

Finally, let us turn to the analysis of the efficiency of this algorithm. The spectrum decomposition in the first step takes polynomial time in d, so is the same with calculation of $|\psi_\alpha\rangle$. The generation of the ϵ-net of the unit ball is standard and can be done in $O((1+\frac{2}{\varepsilon})^{|\Gamma_\epsilon|})\times\mathrm{poly}(|\Gamma_\epsilon|)$. The last operation, finding the Schmidt decomposition, is equivalent to singular value decompositions, and thus can be done in polynomial time in d as well. Also note $|\Gamma_\epsilon|\le\min\{d^2, \|Q\|_{\mathrm{F}}^2/\delta^2\}$. To sum up, the total running time of the algorithm is upper bounded by $O((1+\frac{2}{\varepsilon})^{|\Gamma_\epsilon|})\times\mathrm{poly}(d)$, or equivalently $\exp(O(\log(d)+\delta^{-2}\|Q\|_{\mathrm{F}}^2\ln(\|Q\|_{\mathrm{F}}/\delta)))$.

□

Remarks. One can also apply the observation in the introduction to parallelize the computation in this case. However, the size of the ϵ-net here will depend on some parameter (i.e. $\|Q\|_F/\delta$) other than the input.

Acknowledgement

We thank Zhengfeng Ji and John Watrous for helpful discussions. This research was supported in part by National Basic Research Program of China Awards 2011CBA00300 and 2011CBA00301, and by NSF of United States Award 1017335.

References

[ABD+09] S. Aaronson, S. Beigi, A. Drucker, B. Fefferman and P. Shor, The Power of Unentanglement. *Theory of Computing*, 5, pp. 1-42, 2009.

[AHK05] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta algorithm and applications, 2005.

[AGIK09] D. Aharonov, D. Gottesman, S. Irani, and J. Kempe. The power of quantum systems on a line. *Communications in Mathematical Physics*, 287(1):41–65, 2009.

[Bei10] S. Beigi. NP vs QMAlog(2). *Quantum Information and Computation*, 54(1&2):0141–0151, 2010.

[Bor77] Allan Borodin. On relating time and space to size and depth. *SIAM Journal on Computing*, 6(4):733–744, 1977.

[BT09] H. Blier and A. Tapp, All languages in NP have very short quantum proofs. *Proceedings of the ICQNM*, pp. 34-37, 2009.

[Bra08] F. G. S. L. Brandão. *Entanglement Theory and the Quantum Simulation of Many-Body Physics*. PhD thesis, Imperial College, 2008.

[BCY11] F. G. S. L. Brandão, M. Christandl, and J. Yard. A quasipolynomial-time algorithm for the quantum separability problem. *Proceedings of the 43rd annual ACM Symposium on Theory of Computation (STOC'11)* , pp. 343, 2011.

[CD10] J. Chen and A. Drucker. Short multi-prover quantum proofs for SAT without entangled measurements. arXiv:1011.0716, 2010.

[CF11] A. Chiesa and M. Forbes. Improved Soundness for QMA with Multiple Provers. arXiv:1108.2098, 2011.

[CS11] A. Chailloux and O. Sattath The Complexity of the Separable Hamiltonian Problem. arXiv:1111.5247.

[DF92] A. Defant and K. Floret. *Tensor norms and operator ideals*, North Holland, 1992.

[DPS04] A. C. Doherty, P. A. Parrilo, and F. M. Spedalieri. A complete family of separability criteria. *Phys. Rev. A*, 69:022308, 2004.

[Fan53] K. Fan. Minimax theorems. *Proceedings of the National Academy of Sciences*, 39:42–47, 1953.

[Gat93] J. von zur Gathen. Parallel linear algebra. In J. Reif, editor, *Synthesis of Parallel Algorithms*, chapter 13. Morgan Kaufmann Publishers, Inc., 1993.

[Gha10] S. Gharibian. Strong NP-hardness of the quantum separability problem. *Quantum Information and Computation* , 10:343, 2010.

[GLS93] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*. Springer-Verlag, 1993.

[GNN11] F. L. Gall, S. Nakagawa and H. Nishimura. On QMA Protocols with Two Short Quantum Proofs. arXiv:1108.4306 , 2011.

[Gur03] L. Gurvits. Classical complexity and quantum entanglement. *Journal of Computer and System Sciences*, 69:448, 2004.

[GW10] G. Gutoski and X. Wu. Parallel approximation of min-max problems with applications to classical and quantum zero-sum games. arXiv:1011.2787v2.

[HHH+09] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki. Quantum entanglement. *Rev. Mod. Phys.*, 81:865, 2009

[HLZ10] S. He, Z. Li and S. Zhang. Approximation Algorithms for Homogeneous Polynomial Optimization with Quadratic Constraints. *Mathematical Programming*, Series B 125(2), 353–383, 2010.

[HM10] A. Harrow and A. Montanaro. An efficient test for product states, with applications to quantum Merlin-Arthur games. *Proceedings of IEEE 51st Annual Symposium on Foundations of Computer Science (FOCS'10)*, p. 633, 2010.

[Ioa07] L. M. Ioannou. Computational complexity of the quantum separability problem. *Quantum Information and Computation*, 7:335, 2007.

[IP01] R. Impagliazzo and R. Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(367), 2001.

[JW09] R. Jain and J. Watrous. Parallel approximation of non-interactive zero-sum quantum games. In *Proceedings of the 24th IEEE Conference on Computational Complexity*, pages 243–253, 2009.

[Kal07] S. Kale. *Efficient algorithms using the multiplicative weights update method*. PhD thesis, Princeton University, 2007.

[deK08] E. de Klerk. The Complexity of Optimizing over a Simplex, Hypercube or Sphere: a Short Survey. *Central European Journal of Operations Research* 16(2), 111–125, 2008.

[KMY01] H. Kobayashi, K. Matsumoto and T. Yamakami, Quantum Certificate Verification: Single versus Multiple Quantum Certificates, quant-ph/0110006.

[KMY03] H. Kobayashi, K. Matsumoto and T. Yamakami, Quantum Merlin-Arthur Proof Systems: Are Multiple Merlins More Helpful to Arthur?, *Lecture Notes in Computer Science*, vol. 2906, pp. 189-198, 2003.

[KSV02] A. Kitaev, A. Shen, M. Vyalyi, *Classical and Quantum Computation*. American Mathematical Society, 2002.

[KKR06] J. Kempe, A. Kitaev, and O. Regev. The complexity of the local Hamiltonian problem. *SIAM Journal on Computing*, 35(5):1070–1097, 2006.

[LQNY09] C. Ling, L. Qi, J. Nie and Y. Ye, Bi-Quadratic Optimization over Unit Spheres and Semidefinite Programming Relaxations. *SIAM Journal on Optimization*, Vol. 20, No. 3, pp.1286–1310, 2009.

[MW05] C. Marriott and J. Watrous, Quantum Arthur-Merlin Games, *Computational Complexity*, 14(2): 122-152, 2005

[MWW09] W. Matthews, S. Wehner, and A. Winter. Distinguishability of quantum states under restricted families of measurements with an application to quantum data hiding. *Comm. Math. Phys.*, 291, 2009

[NC00] Michael Nielsen and Issac Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[OT08] R. Oliveira and B. Terhal. The complexity of quantum spin systems on a twodimensional square lattice. *Quantum Information and Computation*, 8(10):0900–0924, 2008.

[RvL11] S. Ragnarsson and C. F. Van Loan Block Tensor Unfoldings. arXiv:1101.2005, 2011.

[So11] A. M. So. Deterministic Approximation Algorithms for Sphere Constrained Homogeneous Polynomial Optimization Problems. *Mathematical Programming*, to appear 2011.

[vN28] J. von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100:295–320, 1928.

[Wat08] J. Watrous. *Lecture Notes on Theory of Quantum Information*. 2008.

[WK06] M. Warmuth and D. Kuzmin. Online variance minimization. In *Proceedings of the 19th Annual Conference on Learning Theory*, volume 4005 of *Lecture Notes in Computer Science*, pages 514–528. Springer, 2006.

[Wu10a] X. Wu. Equilibrium value method for the proof of QIP=PSPACE. arXiv:1004.0264v4 [quant-ph], 2010.

[Wu10b] X. Wu. Parallized solutions to semidefinite programmings in quantum complexity theory. arXiv:1009.2211v3 [quant-ph], 2010.

1. Let $\gamma = \frac{\epsilon}{8Mw}$ and $T = \left\lceil \frac{\ln d}{\gamma^2} \right\rceil$. Also let $W^{(1)} = \mathbb{1}_{\mathcal{X}}$, $d = \dim(\mathcal{X})$.
2. Repeat for each $t = 1, \ldots, T$:
   (a) Let $\rho^{(t)} = W^{(t)} / \operatorname{Tr} W^{(t)}$ and compute $\vec{q}(\rho^{(t)})$ (Equ. (1)). One can then rewrite the vector $\vec{p} - \vec{q}(\rho^{(t)})$ in the polar form $(c_1^{(t)} e^{i\phi_1^{(t)}}, c_2^{(t)} e^{i\phi_2^{(t)}}, \cdots, c_M^{(t)} e^{i\phi_M^{(t)}})$ and choose $\vec{z}^{(t)} = (e^{-i\phi_1^{(t)}}, e^{-i\phi_2^{(t)}}, \cdots, e^{-i\phi_M^{(t)}})$. It is not hard to see such $\vec{z}^{(t)}$ maximizes $\operatorname{Re}\left\langle \vec{p} - \vec{q}(\rho^{(t)}), \vec{z} \right\rangle$.
   (b) Choose $N^{(t)}$ to be
   $$N^{(t)} = \operatorname{Re}\left\langle \vec{p}, \vec{z}^{(t)} \right\rangle \mathbb{1}_{\mathcal{X}} - \frac{1}{2}(Q^{(t)} + Q^{(t)*}) + 2Mw\mathbb{1}_{\mathcal{X}},$$
   where $Q^{(t)} = \sum_{i=1}^{M} e^{+i\phi_i^{(t)}} Q_i$.
   (c) Update the weight matrix as follows: $W^{(t+1)} = \exp(-\gamma \sum_{\tau=1}^{t} N^{(\tau)})$.
3. Return $\tilde{d} = \frac{1}{T} \sum_{t=1}^{T} \left\langle \rho^{(t)}, N^{(t)} - 2Mw\mathbb{1}_{\mathcal{X}} \right\rangle$.

Figure 3: An algorithm that approximates the $d(\vec{p})$ with additive error ϵ.

A Proof of Lemma 3

Theorem 9 (Multiplicative weights update method—see Ref. [Kal07, Theorem 10])**.** *Fix $\gamma \in (0, 1/2)$. Let $N^{(1)}, \ldots, N^{(T)}$ be arbitrary $d \times d$ "loss" matrices with $0 \preceq N^{(t)} \preceq \alpha I$. Let $W^{(1)}, \ldots, W^{(T)}$ be $d \times d$ "weight" matrices given by*

$$W^{(1)} = I \qquad\qquad W^{(t+1)} = \exp(-\gamma(N^{(1)} + \cdots + N^{(t)})).$$

Let $\rho^{(1)}, \ldots, \rho^{(T)}$ be density operators obtained by normalizing each $W^{(1)}, \ldots, W^{(T)}$ so that $\rho^{(t)} = W^{(t)} / \operatorname{Tr} W^{(t)}$. For all density operators ρ it holds that

$$\frac{1}{T} \sum_{t=1}^{T} \left\langle \rho^{(t)}, N^{(t)} \right\rangle \leq \left\langle \rho, \frac{1}{T} \sum_{t=1}^{T} N^{(t)} \right\rangle + \alpha(\gamma + \frac{\ln d}{\gamma T}).$$

Note that Theorem 9 holds for *all* choices of loss matrices $N^{(1)}, \ldots, N^{(T)}$, including those for which each $N^{(t)}$ is chosen adversarially based upon $W^{(1)}, \ldots, W^{(t)}$. This adaptive selection of loss matrices is typical in implementations of the MMW. Consider the algorithm shown in Fig. 3.

Lemma 10 (Restated Lemma 3)**.** *Given any point $\vec{p} \in$ Raw-(M, w) and $\epsilon > 0$, the algorithm in Fig. 3 approximates $dis(\vec{p})$ with additive error ϵ. Namely, the return value $\tilde{d}$ of this algorithm satisfies*

$$\tilde{d} - \epsilon \leq dis(\vec{p}) \leq \tilde{d} + \epsilon.$$

Moreover, the algorithm runs in $poly(d, M, w, 1/\epsilon)$ time. Furthermore, if d is considered as the input size and $M, w, 1/\epsilon \in O(poly\text{-}log(d))$, this algorithm is also efficient in parallel, namely, inside NC.

Proof. The algorithm is a typical application of the matrix multiplicative weight update method. In order to make use of Theorem 9, we need first to show $N^{(t)}$ is bounded for each t. Since $\vec{p} \in \text{Raw-}(M, w)$ and $\|\vec{z}^{(t)}\|_\infty \leq 1$, by Cauchy-Schwartz inequality we have

$$|\text{Re}\left\langle \vec{p}, \vec{z}^{(t)} \right\rangle| \leq \|\vec{p}\|_1 \|\vec{z}^{(t)}\|_\infty \leq M\|\vec{p}\|_\infty \|\vec{z}^{(t)}\|_\infty = Mw.$$

Furthermore we have

$$\|Q\|_{\text{op}} = \|\sum_{i=1}^{M} e^{-i\phi_i^{(t)}} Q_i\|_{\text{op}} \leq \sum_{i=1}^{M} \|Q_i\|_{\text{op}} \leq Mw.$$

Thus by triangle inequality, one can easily find

$$0 \preceq N^{(t)} \preceq 4Mw\mathbb{1}_{\mathcal{X}}.$$

Then we can make use of Theorem 9. Immediately, for any $\rho \in \mathrm{D}(\mathcal{X})$, we have

$$\frac{1}{T}\sum_{t=1}^{T} \left\langle \rho^{(t)}, N^{(t)} \right\rangle \leq \left\langle \rho, \frac{1}{T}\sum_{t=1}^{T} N^{(t)} \right\rangle + \alpha(\gamma + \frac{\ln d}{\gamma T}).$$

Substitute $\alpha = 4Mw, \gamma = \frac{\epsilon}{8Mw}$ and $T = \left\lceil \frac{\ln d}{\gamma^2} \right\rceil$. Also consider the identity $\left\langle \rho^{(t)}, N^{(t)} - 2Mw\mathbb{1}_{\mathcal{X}} \right\rangle = \text{Re}\left\langle \vec{p} - \vec{q}(\rho^{(t)}), \vec{z}^{(t)} \right\rangle$. Then we have for any $\rho \in \mathrm{D}(\mathcal{X})$,

$$\tilde{d} = \frac{1}{T}\sum_{t=1}^{T} \left\langle \rho^{(t)}, N^{(t)} - 2Mw\mathbb{1}_{\mathcal{X}} \right\rangle \leq \text{Re}\left\langle \vec{p} - \vec{q}(\rho), \frac{1}{T}\sum_{t=1}^{T} \vec{z}^{(t)} \right\rangle + \epsilon. \tag{5}$$

Consider the equilibrium value form of $\text{dis}(\vec{p})$ in Equ. (2). For each $\rho^{(t)}$, we always find the $\vec{z}^{(t)}$ that maximizes $\text{Re}\left\langle \vec{p} - \vec{q}(\rho^{(t)}), \vec{z} \right\rangle$. Hence, $\text{dis}(\vec{p}) \leq \tilde{d}$. Let $\rho^\star$ be any equilibrium point of the equilibrium value in Equ. (2). By substituting such $\rho^\star$ into Equ. (5) we have

$$\tilde{d} \leq \text{Re}\left\langle \vec{p} - \vec{q}(\rho^\star), \frac{1}{T}\sum_{t=1}^{T} \vec{z}^{(t)} \right\rangle + \epsilon \leq \text{dis}(\vec{p}) + \epsilon.$$

So far we complete the proof of the correctness of this algorithm. Note that each step in the algorithm only contains fundamental operations of matrices and vectors, which can be done in polynomial time in M, d. Also there are totally $O(T) = \text{poly}(\ln d, M, w, 1/\epsilon)$ steps, thus the whole algorithm can be executed in $\text{poly}(d, M, w, 1/\epsilon)$ time. Moreover, given the fact that fundamental operations of matrices and vectors also admit efficient algorithms in parallel (i.e., NC algorithm), one can easily compose these NC circuits of each step and obtain a NC algorithm as a whole if the total number of steps T is not too large. Precisely, if $M, w, 1/\epsilon \in O(\text{poly-log}(d))$ and d is considered as the input size, this algorithm is also efficient in parallel. □